OAD
Ref③

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 01-237726
(43)Date of publication of application : 22.09.1989

(51)Int.Cl. G06F 9/06
G06F 9/44

(21)Application number : 63-063295
(22)Date of filing : 18.03.1988

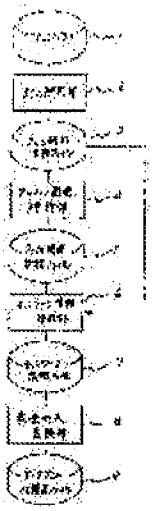
(71)Applicant : HITACHI LTD
(72)Inventor : ITOU KISHIO

(54) AUTOMATIC GENERATING SYSTEM FOR SOFTWARE SPECIFICATION

(57)Abstract:

PURPOSE: To allow the maintenance work of a software to be efficient by analyzing plural software products, preparing the specifying information of a high- order specification and converting these specifying information to the formation of the high-order specification.

CONSTITUTION: A JCL (job control language) analyzing part 2 analyzes a JCL from a JCL library 1 and outputs information concerning an input/output file to a JCL analyzing information file 3. Next, a file relation analyzing part 4 inputs the file 3 and outputs the names of a sharing file and a sharing job to a file relating information file 5. Then, a network information preparing part 6 prepares a network, which expresses connecting relation between a job file and a slip, from the information 3 and file 5 and outputs the network to a network information file 7. Finally, an expressing form converting part 8 inputs the file 7 and converts the file to a data flow specification, which is a purpose. Namely, network information is inputted and an arranging position in the specification, which is prepared concerning respective nodes and branches, is determined.



OA①
Ref③

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平1－237726

⑤Int.Cl.⁴		識別記号	庁内整理番号	⑬公開　平成1年(1989)9月22日
G 06 F	9/06	320	B－7361－5B	
	9/44	320	E－8724－5B	

審査請求　未請求　請求項の数 7　（全8頁）

⑤④発明の名称　ソフトウエア仕様書の自動生成システム

⑳特　願　昭63－63295
㉒出　願　昭63(1988)3月18日

⑦②発明者　伊藤　岸男　東京都品川区南大井6丁目23番15号　株式会社日立製作所大森ソフトウエア工場内
⑦①出願人　株式会社日立製作所　東京都千代田区神田駿河台4丁目6番地
⑦④代理人　弁理士　小川　勝男　外1名

明　細　書

1．発明の名称

ソフトウェア仕様書の自動生成システム

2．特許請求の範囲

1．ソフトウェア仕様を作成する情報処理システムにおいて、ソフトウェア開発において上位仕様書を詳細化することにより得られた少なくとも一つのソフトウェア生産物（プログラム、ジョブ制御文、下位仕様書等）の記述内容を読み取り上位仕様書から引き継がれている内容を抽出し、抽出内容の内同一項目の記述内容をそれぞれ一項目に集約し、一つの項目に集約した項目と単独で存在する項目をまとめネットワーク状の項目情報を作成し、ネットワーク状に配置された各項目の表現形式を変換し目的とする仕様書を生成することを特徴とするソフトウェア仕様書の生成システム。

2．ソフトウェア生産物より上位仕様書から引き継がれている内容の抽出において、利用者が下位仕様書の情報を補う手段を有することを特徴とする特許請求の範囲第1項もしくは第2項記載のソフトウェア仕様書の自動生成システム。

3．ソフトウェア仕様書を生成するために充分な情報が得られなかった場合、複数種類のソフトウェア生産物からの情報を抽出する手段を有することを特徴とする特許請求の範囲第1項、第2項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

4．ソフトウェア生産物がジョブ制御文（以下JCLと呼ぶ）であり、上位仕様書としてデータフローダイアグラムを記述したデータフロー仕様書を生成することを特徴とする特許請求の範囲第1項、第2項、第3項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

5．ソフトウェア生産物がジョブ内のプログラム実行順序及びプログラムの入出力ファイルを図式的に記述したシステムフロー仕様書であり、上位仕様書としてデータフローダイアグラムを記述したデータフロー仕様書を生成することを特徴とする特許請求の範囲第1項、第2項、第

3項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

6．ソフトウェア生産物がデータベース記述言語によって記述されたデータベース定義情報であり、上位仕様書としてデータベース内の構成要素の関連を表すE．R（Entity,Relation）仕様書を生成することを特徴とする特許請求の範囲第1項、第2項、第3項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

7．ソフトウェア生産物がオンラインプログラムのソースプログラムであり、上位仕様書として画面名称と画面遷移状態を表す画面遷移仕様書を生成することを特徴とする特許請求の範囲第1項、第2項、第3項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

3．発明の詳細な説明

〔産業上の利用分野〕

本発明は、ソフトウェア開発の計算機による支援に係り、特にソフトウェアの保守に好適なソフトウェア仕様書の作成システムに関する。

〔従来の技術〕

一般にソフトウェアの開発では、開発すべきソフトウェアの具備すべき機能を定め、その機能を実現するソフトウェアを段階的に詳細化して設計を進め、最終的にプログラムを作成する方法が用いられる。

設計結果は、詳細化の各階段ごとに設計仕様書として記述され、次の設計階段では、前の段階で作成された仕様書の各要素を詳細化した仕様書が作成される場合が一般的である。

これらの仕様書やプログラムの作成を効率化するために各種の仕様書のエディタや、プログラムエディタなどの開発支援ツールが開発されている。

これらの方法については例えば「日経コンピュータ」1986年7月7日号P121～P136「システムフローを自動生成するソフトEAGLE2」に見られる通りである。

〔発明が解決しようとする課題〕

従来、新規にソフトウェアを開発する場合には有効であるが、これを用いて開発したソフトウェアの保守を行うには次の問題点が残されている。初期の段階で作成された上位仕様書と、これを詳細化して作成されたソフトウェア生産物（プログラム、ジョブ制御文、下位仕様書）と内容が一致しない場合が生じる。この場合、仕様書は実際のプログラムやジョブ制御文（以下JCLと呼ぶ）の内容を表わさなくなり、保守が困難になる。

本発明の目的は、仕様書とプログラム、あるいは上位仕様書とこれを詳細化した仕様書の不一致を無くすことにより、ソフトウェアの保守作業を容易にすることにある。

〔課題を解決するための手段〕

上記目的は、複数の詳細化された仕様書あるいはプログラムやJCLなどのソフトウェア生産物を解析してその中の上位仕様書に含むべき仕様情報と、抽出情報の内複数の仕様書の内で共通に含まれる仕様情報を統合し、統合されない抽出情報を含めて上位仕様書の仕様情報を作成し、この仕様情報を上位仕様書の形式に変換することにより達成される。

もし、一種類のソフトウェア生産物内の仕様情報のみでは上位仕様書を作成するために必要な情報が得られない場合、他の下位ソフトウェア生産物の情報を解析しその中の情報を前記抽出情報に付加する手段を付加することにより達成できる。

また、ソフトウェア生産物内の情報のみでは上位仕様情報を作成できない場合、下位の仕様情報に対し利用者が上位仕様書を作成するために必要な情報を加えることにより達成できる。

〔作　用〕

上記手段により、下位の仕様書やプログラム等のソフトウェア生産物より、その内容と一致する上位仕様書を効率よく作成することができる。

これにより、仕様書間あるいは仕様書とプログラム間の不整合をなくすことができるため、ソフトウェア保守作業におけるソフトウェアの理解を効率よく行うことができる。

〔実施例〕

以下、本発明の一実施例を図を用いて説明する。

本実施例は、計算機上での処理単位となるジョ

ブ内での(1)プログラムの実行順序及び(2)プログラムに対する入出力装置を記述するジョブ制御文（以下ＪＣＬと呼ぶ）からジョブ間のデータの流れを表す仕様書であるデータフロー仕様書を生成する方法について述べる。データフロー仕様書はＪＣＬの上位仕様書であり、公知のデータフローダイアグラムで表現する。

第１図は、本発明の一実施例を実現する機能ブロック図を示す図である。第２図は、第１図の実施例を実現する装置のハードウェア装置の構成図を示し、ＣＰＵ１０１、メモリ１０２、外部記憶装置１０３、入力装置１０４および表示装置１０５からなる。

第１図のＪＣＬライブラリ１は、データフロー仕様書の生成に使用するＪＣＬ群を納めたファイルを示す。ＪＣＬ解析部２はＪＣＬライブラリー１よりＪＣＬ群を入力し、入力した各ＪＣＬについて下記規則によりＪＣＬを解析し、その中の当該ジョブの入出力ファイルに関する情報をＪＣＬ解析情報ファイル３へ出力する。

ＪＣＬ解析規則は、次の通りである。

(1) 入出力装置のハードウェア種別としてプリンターが割り当てられている場合には、帳票出力として解析情報を出力する。

(2) 入出力装置のハードウェア種別としてプリンター以外が割り当てられている場合には、ファイル名称と入出力区分を解析情報として出力する。

第３図はＪＣＬライブラリー１に蓄積されているＪＣＬの例であり、第４図は第３図のＪＣＬを解析してＪＣＬ情報解析ファイル３に蓄積されるＪＣＬ解析情報例である。

第３図に示すＪＣＬ１では、１０で示すＪＯＢ文にて"ＪＯＢ１"というジョブ名称があたえられている。１１から１３に示すＡＳＮ文により当該ジョブが使用する入出力ファイルの記述がなされている。ＡＳＮ文１１では、入出力装置の種別を記述するＥＱＵＩＰオペランド１６にて、磁気ディスク装置を表す"ＤＩＳＫ"という記述がされており、この磁気ディスク装置上のファイルに作成されているファイルのファイル名称は１７に示すＦＩＬＥオペランドにより"ＦＩＬＥ１"と定義されている。そして１８に示すＩ／Ｏオペランドにて当該ファイルが、本プログラムに対し入力ファイルであるか出力ファイルであるかが示されている。本例の場合このＩ／Ｏオペランドにて入力ファイルであることを示す"ＩＮＰＵＴ"が記述されている。ＡＳＮ文１２も同様に入出力装置として磁気ディスク装置が割り当てられており、そのファイル名称が"ＦＩＬＥ２"であるが、ファイルの入出力種別を記述するＩ／Ｏオペランド１９に出力ファイルであることを示す"ＯＵＴＰＵＴ"が記述されている。ＡＳＮ文１３では、入出力装置の種別を記述するＥＱＵＩＰオペランド２０にてプリンター装置を表す"ＰＲＩＮＴＥＲ"という記述がされており、帳票名称２１に示すＯＵＴＣＬＡＳＳオペランドにて"Ａ"と定義されている。文１４は、当該ジョブにて実行するプログラムの名称を記述する文であり、本例では、"ＰＲＯＧＲＡＭ１"というプログラムが実行される。文１５は、ジョブの終端を示す。

同様の解析を第３図のＪＣＬ２について施し、出力したＪＣＬ解析情報ファイル例が第４図である。

次に、第１図のファイル関連解析部４は、ＪＣＬ解析情報ファイル３を入力し、複数のジョブ間で同一ファイル名称を利用しているものを抽出し、共用ファイル名称及び、共用を行なうジョブのジョブ名称をファイル関連情報ファイル５へ出力する。

第５図は第４図のＪＣＬ解析情報ファイル５を入力した場合の共用ファイル情報の抽出例である。第５図のＦＩＬＥ１　２２については、第４図のＪＣＬ解析情報ファイルでは、ＪＯＢ１の入力ファイルのみに利用されているため、当該ファイルの参照ジョブ名称として"ＪＯＢ１"が登録される。第５図のＦＩＬＥ２　２３については、第４図よりジョブ名"ＪＯＢ１"で更新され、ジョブ名"ＪＯＢ２"で参照されるため、更新ジョブとして"ＪＯＢ１"、参照ジョブとして"ＪＯＢ２"が登録さ

れている。同様に、第５図のＦＩＬＥ３　２４は、“ＪＯＢ２”の出力ファイルとしてのみ利用されているため、更新ジョブ名称として“ＪＯＢ２”が登録されている。帳票出力２５は、ジョブ名“ＪＯＢ１”より出力されていることが第４図のＪＣＬ解析情報より判明するため更新ジョブ名称として“ＪＯＢ１”が登録されている。

次に第１図のネットワーク情報作成部６は、ＪＣＬ解析情報３とファイル関連情報ファイル５よりジョブ、ファイル及び帳票の接続関係を表すネットワーク情報を作成し、ネットワーク情報ファイル７へ出力する。ここで作成するネットワーク情報は、ＪＣＬ解析情報中に存在する全てのジョブ、ファイル及び帳票をノードとし、ファイルとジョブまたは、ジョブと帳票間の参照関係に依存するブランチを持つものとする。また、ファイル関連情報解析部により判明された複数ジョブ間で共用されているファイルに対しては、当該ファイルに関するノードが一つだけ存在するネットワーク情報を作成する。第４図に示すＪＣＬ解析情報と第５図に示すファイル関連情報により作成されるネットワーク情報を図式的に表現した例を第６図に示す。第６図中の丸付文字列、枠付文字列及び二重枠付文字列は、各々ジョブ、ファイル、帳票のノードとその名称を意味する。また、矢印は、ジョブとファイルまたは、ジョブと帳票間のブランチを示し、矢印の向きが参照方向を表す。第６図のＪＯＢ１　２６に着目した場合、第４図のＪＣＬ解析情報より、入力ファイルとして“ＦＩＬＥ１”、出力ファイル及び帳票として“ＦＩＬＥ２”及び“Ａ”が与えられているため、ジョブを表すノード“ＪＯＢ１”への入力となるノードとして“ＦＩＬＥ１”、ノード“ＪＯＢ１”からの出力となるノードとしてファイル“ＦＩＬＥ２”及び帳票“Ａ”が接続されている。“ＦＩＬＥ２”２７に着目した場合、第４図によればジョブ“ＪＯＢ１”と“ＪＯＢ２”で共用しているファイルであるが、ネットワーク情報上は、当該ファイルを示すノードが一つだけ存在していることがわかる。

最後に第１図の表現形式変換部８にて、ネットワーク情報ファイル７を入力し、目的とするデータフロー仕様書に変換する。表現形式変換部８では、ネットワーク情報を入力し、各ノード及びブランチについて作成すべき仕様書内の配置位置を決定する。配置位置決定アルゴリズムは公知であり、例えば情報処理１９８６ＮＯ６“応用指向メモリ”に示されている。配置位置決定後、各ノード及びブランチについて表現形式を変換しデータフロー仕様書ファイル９へ出力する。

第７図にジョブとジョブ間のデータの流れをネットワーク形式で表現するデータフローダイアグラムの一種であるＳＤＦ（Structured Dataflow Diagram）にて表現データフロー仕様書の例を示す。

以上の実施例では、データフロー仕様書を生成するために必要なファイルの入出力に関する情報が全てＪＣＬより得られる場合であるが、本実施例を適用するコンピューターシステムによっては、ＪＣＬのみから十分な情報が得られない場合がある。このような場合に好適な実施例を次に示す。

第８図は本実施例の機能ブロック図を示す。第８図は第１図とほぼ同様であるが、ＪＣＬ解析情報ファイル編集部２８を加えた点が異なる。

第９図は第８図のＪＣＬライブラリ１に蓄積されるＪＣＬの例を示す。第９図のＪＣＬは第３図に示す前実施例とほぼ同様であるが、第９図中の“ＪＯＢ１”実行用ＪＣＬの内、下線２９で示す“ＦＩＬＥ２”に関する入出力区分の記述がなされていない点が異なる。第９図のＪＣＬを第８図のＪＣＬ解析部２で解析した場合のＪＣＬ解析情報ファイル３の例を第１０図に示す。第１０図では、ジョブ名称“ＪＯＢ１”、ファイル名称“ＦＩＬＥ２”に関する入出力区分下線部３０が空欄となっている。このＪＣＬ解析情報だけでは第８図に示すファイル関連解析部４に入力した場合、当該ジョブがファイルを参照するかあるいは、更新するかが不明であるためネットワーク情報を構成することができない。そこで、ＪＣＬ解析情報編集部２８にて、入出力区分を人手で入力することによりネットワーク情報を完成させることができる。ジョブ名称、ファイル名称、帳票名称についての

情報も同様にして追加，削除，更新を可能とする。

以下、第１図の実施例と同様の手順で目的とするデータフロー仕様書を作成する。

上記実施例では、データフロー仕様書生成のため不足する情報をＪＣＬ解析情報を入手により補填することにより補ったが、ＪＣＬライブラリー中に記述されているＪＣＬによって実行される各プログラムの詳細仕様であるプログラムの情報を解析しこの情報により当該不足情報を補う場合の実施例を第１１図に示す。

第１１図の実施例は第１図の実施例とほぼ同様であるが、ソースプログラムライブラリ３１，ソースプログラム解析部３２，ソースプログラム解析情報ファイル３３及びソースプログラム解析情報統合部３４を付加した点が異なる。

第１１図の３４において当該ジョブの使用ファイルの入出力区分の情報を補う。

第１２図は，第９図のＪＣＬ１によって実行されるプログラム“ＰＲＯＧＲＡＭ１”のＣ言語によるコーディング例を示す。第１２図では，fopen文３５から３７により当該プログラムの入出力ファイルの割り当てを行なった後データ処理を行なっている。ここでは，文３５により入力ファイルとして“ＦＩＬＥ１”，文３６により出力ファイルとして“ＦＩＬＥ２”，文３７により帳票出力ファイルとして“Ａ”を割り当てている。本プログラムは，ソースプログラムライブラリ３１に記憶される。

第１１図のソースプログラム解析部３２では，ソースプログラムライブラリ３１より本プログラムを入力し文３５から３７で記述されているプログラムの入出力に関する情報を第１１図のソースプログラム解析情報ファイルへ３３へ出力する。

第１２図のプログラム例を解析して得られたソースプログラム解析情報ファイル３３内の情報例を第１３図に示す。次に第１１図ソースプログラム解析情報統合部３４によりＪＣＬ解析情報ファイル３とソースプログラム解析情報ファイル３３を入力し，各々のファイル中のプログラム名称とファイル名称を基に突合せを行ない，ＪＣＬ解析情報ファイル中の入出力区分がないファイルに対しソースプログラム解析情報ファイル３３より得られた入出力区分の情報を付加する。

第１４図の場合ＪＣＬ解析情報ファイルのプログラム名称“ＰＲＯＧＲＡＭ１”，ファイル名称“ＦＩＬＥ２”の入出力区分３８が不定であるが，ソースプログラム解析情報ファイルより同一プログラム名称と同一ファイル名称を持つ行３９を検索し，ＪＣＬ解析情報ファイルに当該行の入出力区分の情報を埋めている。

以下第１図の実施例と同様の手順によりデータフロー仕様書を生成することができる。

上記実施例の他にジョブ内のプログラム実行順序及びプログラムの入出力ファイルを図式的に記述したシステムフロー仕様書からデータフロー仕様書を生成、データベース記述言語によって記述されるデータベース定義情報よりデータベース内の構成要素の関連を表すＥ．Ｒ(Entity.Relation)仕様書を生成、オンラインプログラムのソースプログラムよりオンラインプログラムの画面名称と画面遷移状態を表す画面遷移仕様書を生成する事が本方式で実現できる。

〔発明の効果〕

本発明によれば，作成されたソフトウェア生産物（プログラム，ＪＣＬ，下位仕様書等）から上位の仕様書を自動生成できるため，上位仕様書の内容と下位のソフトウェア生産物との整合性が保証されるのでソフトウェア保守作業の効率化が図れる。

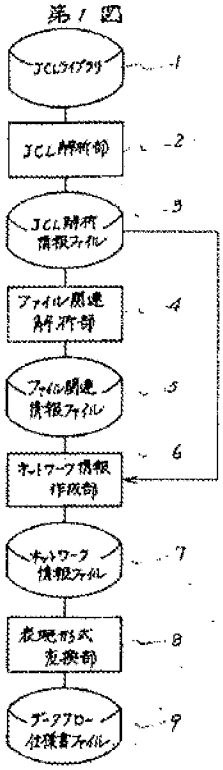
4．図面の簡単な説明

第１図は本発明の一実施例の機能ブロック図，第２図は本発明のハードウェア環境を示す図，第３図は本発明の一実施例の入力となるＪＣＬを示す図，第４図はＪＣＬ解析結果を示す図，第５図はＪＣＬ解析結果をファイル／帳票名称に着目してジョブによる参照／更新関係にまとめ直した図，第６図は第５図のファイル／帳票名称のジョブによる参照／更新関係をネットワーク状に展開し図式的に表した図，第７図は第６図のネットワーク情報の各ノードを形式変換して得られたデータフ

ローダイアグラムを示す図、第８図は本発明の第２図の実施例の機能ブロック図、第９図は第２の実施例の入力となるＪＣＬを示す図、第１０図は第９図のＪＣＬ例の解析結果を示す図、第１１図は本発明の第３の実施例の機能ブロック図、第１２図は入力となるソースプログラムの例を示す図、第１３図はソースプログラムの解析結果を示す図、第１４図はプログラム解析結果により不足情報を補った例を示す図である。

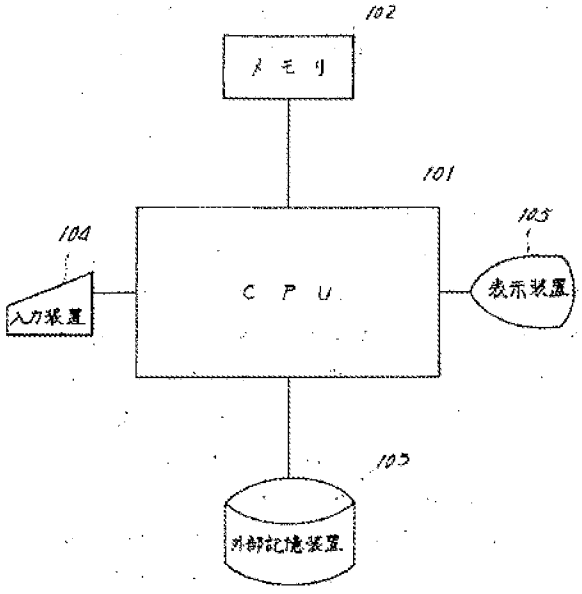
１…ＪＣＬライブラリ，２…ＪＣＬ解析部，３…ＪＣＬ解析情報ファイル，４…ファイル関連解析部，５…ファイル関連ファイル，６…ネットワーク情報作成部，７…ネットワーク情報ファイル，８…表現形式変換部，９…データフロー仕様書ファイル。

第１図



代理人弁理士　小　川　勝　男

第２図



第３図

(a)

＜ＪＣＬ１：ジョブ名"ＪＯＢ１"実行用ＪＣＬ＞

```
10 - JOB   JOB1
11 - ASN   EQUIP=DISK, FILE=FILE1, I/O=INPUT
12 - ASN   EQUIP=DISK, FILE=FILE2, I/O=OUTPUT
13 - ASN   EQUIP=PRINTER, ..., OUTCLASS=A
14 - EXEC  PROGRAM1
15 - END
```

(16, 17, 18, 19, 20, 21)

(b)

＜ＪＣＬ２：ジョブ名"ＪＯＢ２"実行用ＪＣＬ＞

```
JOB   JOB2
ASN   EQUIP=DISK, FILE=FILE2, I/O=INPUT
ASN   EQUIP=DISK, FILE=FILE3, I/O=OUTPUT
EXEC  PROGRAM2
END
```

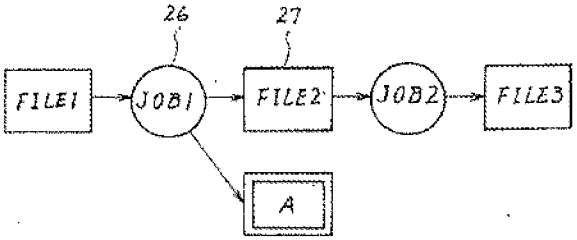
第4図

ジョブ名称	入出力区分	ファイル名称	帳票名称
JOB1	入力	FILE1	
	出力	FILE2	
	出力		A
JOB2	入力	FILE2	
	出力	FILE3	

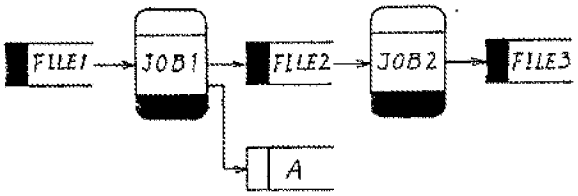
第5図

	ファイル/帳票名称	更新ジョブ名称	参照ジョブ名称
ファイル	FILE1		JOB1 ~22
	FILE2	JOB1	JOB2 ~23
	FILE3	JOB2	~24
帳票	A	JOB1	~25

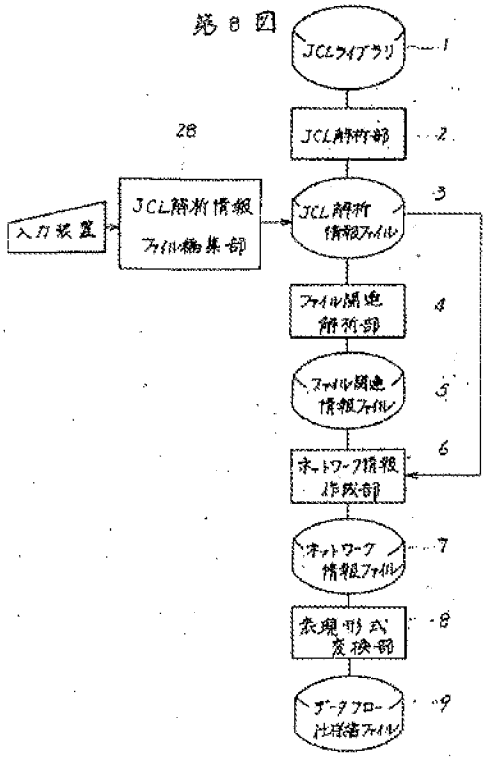
第6図



第7図



第8図



第9図

(a)

< JCL1 : ジョブ名"JOB1" 実行用JCL >

```
JOB   JOB1
ASN   EQUIP=DISK,FILE=FILE1,I/O=INPUT
ASN   EQUIP=DISK,FILE=FILE2          29
ASN   EQUIP=PRINTER,,,OUTCLASS=A
EXEC  PROGRAM1
END
```

(b)

< JCL2 : ジョブ名"JOB2" 実行用JCL >

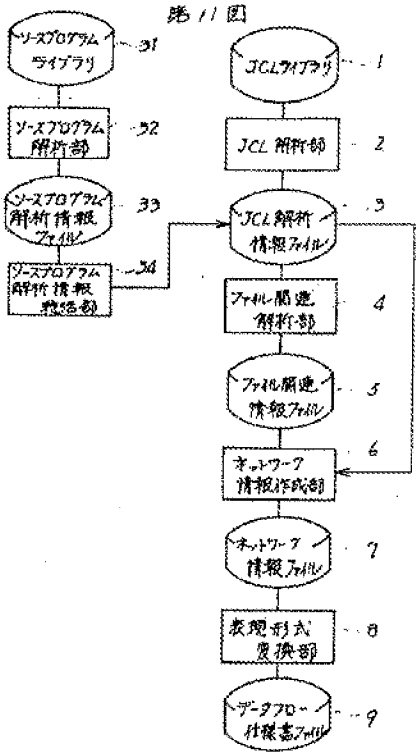
```
JOB   JOB2
ASN   EQUIP=DISK,FILE=FILE2,I/O=INPUT
ASN   EQUIP=DISK,FILE=FILE3,I/O=OUTPUT
EXEC  PROGRAM2
END
```

第10図

ジョブ名称	プログラム名称	入出力区分	ファイル名称	帳票名称
JOB1	PROGRAM1	入力	FILE1	
			FILE2	
		出力		A
JOB2	PROGRAM2	入力	FILE2	
		出力	FILE3	

30

第11図



第12図

< PROGRAM1のソースプログラム >

```
main() {
35 ---fopen("FILE1", input);
36 ---fopen("FILE2", output);
37 ---fopen("A", output);
        データ処理部
}
```

第13図

プログラム名称	入出力区分	ファイル名称	帳票名称
PROGRAM1	入力	FILE1	
	出力	FILE2	
	出力		A

第14図